

## REMARKS

Claims 1-2 are pending.

Claims 1-2 stand rejected.

Claims 1 and 2 have been amended to correct respective antecedent basis issues and not for reasons of patentability.

Claims 2-20 have been added.

## Specification

The specification is objected to because the Abstract exceeds 150 words. The Abstract has been amended to overcome this objection.

## Claim Rejections - 35 U.S.C. § 102

Claims 1-2 stand rejected under 35 U.S.C. § 102(e) as being anticipated by U.S. Patent No. 6,151,643 issued to Cheng et al. (hereinafter "*Cheng*"). Applicants respectfully traverse the rejection.

*Cheng* relates to a:

A system and method [to] update client computers of various end users with software updates for software products installed on the client computers, the software products manufactured (sic) by diverse, unrelated software vendors. The system includes a service provider computer system, a number of client computers and software vendor computer systems communicating on a common network. The service provider computer system stores in an update database information about the software updates of the diverse software vendors, identifying the software products for which software updates are available, their location on the network at the various software vendor computer systems, information for identifying in the client computers the software products stored thereon, and information for determining for such products, which have software

updates available. ... Changes to the client computer during installation are monitored and archived, allowing the updates to be subsequently removed by the user. *Cheng*, Abstract.

Applicants respectfully submit that the Examiner has inconsistently applied teachings of *Cheng*, and, thus, *Cheng* does not teach or suggest the present invention.

**Claim 1.**

The Examiner states that *Cheng*, col. 2, line 62 to col. 3, line 12, teaches “generating a configuration of software components, each having associated information” and, thus, anticipates “A method of generating a configuration comprising a plurality of components each having an associated context and context state” recited by claim 1. *Cheng*, col. 2, line 62 to col. 3, line 12 recites in relevant part:

In accordance with one aspect of the present invention, there is provided a system and method that automatically updates software components from numerous diverse software vendors on the computer systems of a plurality of end users. The system includes at least one database that stores software update information for a plurality of software products manufactured by diverse software vendors. ... The software update information in the database specifies the software update program or files and their network location on the computer system of the software vendors, which computer systems are connected over the network to the service provider computer system. The database further stores information that describes an installation process for installing the software update on a user's computer. (emphasis added).

Thus, the Examiner is equating the “software components” of *Cheng* with the “components” of claim 1 and equating “software update information in the database [that] specifies the software update program or files for a plurality of software products manufactured by diverse software vendors” as the “context state” of claim 1.

The Examiner also states that *Cheng*, col. 10, lines 26-32, teaches “that the information relates to context states having a plurality of values” and, thus, anticipates “said context state equal to one or more of a plurality of values” recited by claim 1. *Cheng*, col. 10, lines 26-32 recites in relevant part:

The update database 709 maintains information identifying a large number of software products, information about the software updates that are available from the diverse software product vendors for these software products, information for identifying software products installed on a client computer 101, and for uniquely distinguishing the versions and names of installed software products. (emphasis added).

Thus, since the Examiner is associating “software update information in the database [that] specifies the software update program or files” with “a context state” ( as established above), the Examiner must be equating “said context state equal to one or more of a plurality of values” of claim 1 with “information about the software updates that are available” of *Cheng*.

The Examiner next states that *Cheng*, col. 8, line 62 to col. 9, line 16, teaches “archiving or caching the current state of the system in response to an installation requirement” and, thus, anticipates “caching a current context state in response to a requirement for the installation of a component” recited by claim 1. *Cheng*, col. 10, lines 26-32 recites in relevant part:

In the preferred embodiment, the client application 104 executes 211 an install monitor prior to actually installing the software update. The install monitor, as further described below, records the changes made to the client computer 101 as a result of the installation of the software update. This information is archived by the install monitor and allows the user to "undo" or remove any number of installations, and restore the client computer 101 to its state prior to each such installation. Accordingly, the client application 104 performs 212 the installation, executing any necessary decompression, installation, or setup applications necessary to install the software update. During the installation process 212 the install monitor records 213 any changes made to the system configuration, including changes to various configuration files, additions or deletions of files,

and additions or deletions of directories. The changes may be recorded in a variety of manners, such as building descriptions of the modifications of the files, or alternatively, storing copies of files prior to their alteration or deletion. Once the installation is complete, the install monitor archives 214 the changes. This process 208 is repeated for each software update to be installed. (emphasis added).

Thus, the Examiner is now associating “the install monitor ... records the changes made to the client computer 101 as a result of the installation of the software update” with “caching a current context state in response to a requirement for the installation of a component” of claim 1.

The Examiner also states that *Cheng*, col. 9, lines 28-55, teaches “restoring an archived or cached state after the installation of the component” and, thus, anticipates “restoring a cached state of the context upon completing installation of the component” recited by claim 1. *Cheng*, col. 9, lines 28-55 recites in relevant part:

At some subsequent point, the user may decide to undo a previous installation, for example, due to dissatisfaction with the software product. The user may use a recovery feature of the client application 104 to undo 216 the installation. ... The recovery function deletes the files installed for the software update, and using the archived information created by the install monitor during the installation of the product, restores the client computer system 101 to its configuration immediately before the installation of the product.

Thus, the Examiner is now associating “undo[ing] a previous installation [of a software update to the client computer]” with “restoring a cached state of the context upon completing installation of the component” of claim 1.

However, Applicants respectfully submit that the Examiner has inconsistently applied the teachings of *Cheng*. More specifically, as established above, the Examiner initially equated “context state” from claim 1 with “software update information that specifies the software update program or files for a plurality of software products manufactured by diverse software vendors”

from *Cheng*, and equated “said context state equal to one or more of a plurality of values” of claim 1 with “information about the software updates that are available.”

Thus, following the Examiner’s arguments and initial association of context states with available software update information for software then-installed on a computer system, “caching a current context state in response to a requirement for the installation of a component” would be equated with ‘caching information about available software updates for software then-installed on the computer system’. However, *Cheng* does not teach ‘caching information about available software updates for software then-installed on the computer system’. *Cheng* teaches recording changes to “the client computer 101 as a result of the installation of the software update.” *Cheng*, col. 10, lines 26-32.

Continuing the Examiner’s arguments and initial association of context states with available software update information for software then-installed on a computer system, “restoring a cached state of the context upon completing installation of the component” or claim 1 would be equated with ‘restoring an archived or cached state of the information about available software updates for software then-installed on the computer system after the installation of the software update.’ However, *Cheng* does not teach ‘restoring an archived or cached state of the information about available software updates for software then-installed on the computer system after the installation of the software update.’. *Cheng* teaches “undo[ing] a previous installation [of a software update to the client computer].” *Cheng*, col. 9, lines 28-30.

Accordingly, Applicants respectfully submit that the Examiner has inconsistently applied the teachings of *Cheng*. In other words, at first the Examiner equated “context state” with ‘information about available updates to installed software on the computer.’ Then the Examiner

equated “context state” with ‘the state of the computer 101 before installation of the software updates.’ Since, ‘information about available updates to installed software on the computer’ is NOT the same as “the state of the computer 101 before installation of the software updates’, the Examiner has inconsistently applied the teachings of *Cheng* to the present invention of claim 1. Thus, *Cheng* cannot teach or suggest the present invention of claim 1.

Table 1 depicts the teachings of claim 1, the Examiner’s interpretation of *Cheng*, and the actual teachings of *Cheng*. The Examiner’s shift in the identification of context states in *Cheng* is apparent from the contrast between rows 2 & 3 and rows 4 & 5.

	Claim 1	Examiner	Cheng
1.	components	software components	software components
2.	[components] each having an context and <u>context state</u>	[software components,] each having associated information	<u>software update information</u> in the database [that] <u>specifies the software update program or files for a plurality of software products</u> manufactured by diverse software vendors.
3.	said <u>context state</u> equal to one or more of a plurality of values	the information relates to context states having a plurality of values	<u>information about the software updates that are available</u>
Identification of “context state” by the Examiner switches between ‘software update information available for installed software’ (ROWS 2 & 3) to ‘the then-current installation of software’ (ROWS 3 & 4).			
4.	caching a current <u>context state</u> in response to a requirement for the installation of a component	archiving or caching the current state of the system in response to an installation requirement	the install monitor ... records the <u>changes made to the client computer 101 as a result of the installation</u> of the software update

	Claim 1	Examiner	Cheng
5.	restoring a cached <u>state of the context</u> upon completing installation of the component	restoring an archived or cached state after the installation of the component	<u>undo[ing] a previous installation</u> [of a software update to the client computer]

TABLE 1

**Claim 2.**

For reasons similar to claim 1, Applicants respectfully submit that *Cheng* does not teach or suggest claim 2. Applicants respectfully submit that the Examiner has inconsistently applied the teachings of *Cheng*. More specifically, the Examiner initially equated “context state” from claim 2 with “software update information that specifies the software update program or files for a plurality of software products manufactured by diverse software vendors” from *Cheng*, and equated “said context state equal to one or more of a plurality of values” of claim 2 with “information about the software updates that are available.”

Thus, following the Examiner’s arguments and initial association of context states with available software update information for software then-installed on a computer system, “a cache configured to hold a current context state in response to a requirement for the installation of a component” would be equated with ‘a cache configured to hold current information about available software updates for software then-installed on the computer system’. However, *Cheng* does not teach ‘a cache configured to hold current information about available software updates for software then-installed on the computer system’. *Cheng* teaches recording changes to “the client computer 101 as a result of the installation of the software update.” *Cheng*, col. 10, lines 26-32.

Continuing the Examiner's arguments and initial association of context states with available software update information for software then-installed on a computer system, "a restore module adapted to restore a cached state of the context upon a completion of an installation of the component" of claim 2 would be equated with 'a restore module adapted to restore information about available software updates for software then-installed on the computer system after the installation of the software update.' However, *Cheng* does not teach 'a restore module adapted to restore information about available software updates for software then-installed on the computer system after the installation of the software update.'. *Cheng* teaches "undo[ing] a previous installation [of a software update to the client computer]." *Cheng*, col. 9, lines 28-30.

Accordingly, Applicants respectfully submit that the Examiner has inconsistently applied the teachings of *Cheng*. In other words, at first the Examiner equated "context state" with 'information about available updates to installed software on the computer.' Then the Examiner equated "context state" with 'the state of the computer 101 before installation of the software updates.' Since, 'information about available updates to installed software on the computer' is NOT the same as "the state of the computer 101 before installation of the software updates", the Examiner has inconsistently applied the teachings of *Cheng* to the present invention of claim 2. Thus, *Cheng* cannot teach or suggest the present invention of claim 2.

Accordingly, Applicants respectfully request withdrawal of the rejection of claims 1 and 2.

#### **New Claims.**



Applicants respectfully submit that each of the new dependent claims is allowable for reasons in addition to those of the claim upon which each dependent claim depends. For example, *Cheng* teaches that:

The client application 104 executes 211 an install monitor prior to actually installing the software update. The install monitor, as further described below, records the changes made to the client computer 101 as a result of the installation of the software update. This information is archived by the install monitor and allows the user to "undo" or remove any number of installations, and restore the client computer 101 to its state prior to each such installation. *Cheng*, col. 10, lines 26-32.

In contrast to the teachings of *Cheng*, the present invention of claim 3 recites “restoring a cached state of the context upon completing installation of the component while retaining installation of the component as part of the configuration.” (emphasis added).

In contrast to the teachings of *Cheng*, the present invention of claim 10 recites “wherein the restore module is further adapted to retain installation of the component as part of the configuration after any restoration of the cached state.”

The present invention of new claims 5 and 12 recites wherein the context associated with a component represents a limited set of additional components that are compatible as additions to a particular configuration with the associated component.” The present invention of new claims 6 and 13 recites “wherein the context associated with a component represents a class of components that are compatible as additions to a particular configuration with the associated component.” *Cheng* only refers to software updates, which are not additions.

*Cheng* neither teaches nor suggests the present invention of claims 7 and 14 which recites “wherein each component is associated with a context attribute that allows identification of the

context of each component, the method further comprising processing the context attribute associated with the installed component to determine the context associated with the installed component.”

Additionally, *Cheng* neither teaches nor suggests the present invention of new claim 17, which recites:

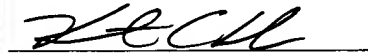
A method of modeling a product to facilitate heterogeneous configuration of the product, the method comprising:

- representing a plurality of components in a configuration model of the product, wherein at least of a subset of the components are included in at least two classes of compatible components and each class of compatible components represents a separate context;
- associating a context attribute with each components in the subset of components, wherein the context attribute identifies each context that includes the component associated with the context attribute; and
- a constraint for each class of compatible components that requires determination of the context of each component installed in a configuration of the product.

## CONCLUSION

In view of the amendments and remarks set forth herein, the application is believed to be in condition for allowance and a notice to that effect is solicited. Nonetheless, should any issues remain that might be subject to resolution through a telephonic interview, the Examiner is requested to telephone the undersigned.

I hereby certify that this correspondence is being deposited with the United States Postal Service as First Class Mail in an envelope addressed to: Mail Stop Amendment, Commissioner for Patents, PO Box 1450, Alexandria, VA 22313-1450, on February 28, 2005.



Attorney for Applicant(s)

2-28-2005

Date of Signature

Respectfully submitted,



Kent B. Chambers  
Attorney for Applicant(s)  
Reg. No. 38,839